Mint for Old English Pounds

Dennis Shasha

Omniheurist Course

Computer Science

Description

You are in charge of designing the denominations of coins. However, we're going to do this for old English pounds which were worth 240 pence. You have made a study of the subpound prices and have determined that each multiple of 5 cents price is N times as likely as a non-multiple of 5 cents. For example, if N = 4, then 15 cents is four times as likely to be the price as 43 cents. But any 5 cent multiple is equally likely as any other 5 cent multiple and ditto for the non-5 cent multiples. The N will be given in class and your programs will have 2 minutes to solve each of the following two problems. (N will be 1 or greater but may not be an integer.)

- Your first job is to design a set of 7 coin denominations such that the expected number of coins required to give exact change for a purchase is minimized given these constraints. This is called the Exact Change Number. Using the U.S. denominations, the Exact Change Number for 43 cents can be realized by one quarter, one dime, one nickel, and three pennies, thus giving a total of 6.
- Let the Exchange Number of a purchase be the number of coins given from the buyer to the seller plus the number given in change to the buyer from the seller. For an item costing 43 cents using the U.S. denominations, the Exchange Number can be realized by having the buyer pay 50 cents and receiving a nickel and two pennies in return, giving a total of only 4. You can assume the availability of 1 pound. So, the Exchange Number for 239 cents is 1 since a penny is returned after handing the seller 1 pound. Your second job is to design a set of 7 coin denominations such that the Exchange Number of coins required for a purchase is minimized given these constraints.
- For both jobs, please print out the score and the denominations for each price paid. Note that the denominations may differ for the two problems (exact change tends to use larger combination).

Hint: dynamic programming is a good start. You can use either one dimensional dynamic programming or two dimensional dynamic programming in the spirit of string matching. That is the inner loop. As for the outer loop, you might have to think whether any coin sizes are impossible. You might also find it useful to evaluate the cost on multiples of 5s separately from the costs on non-multiples of 5.

Scoring: Score is sum of the costs of all non-multiples of 5 + sum of N * costs of the multiples of 5. For example, if the cost of every entry were 2, then the number of multiples of 5 cents would be 235/5 = 47. Then the total score would be (239-47)*2 + (47*N*2). Please print out the score and the denominations.